

基于3D-LBM的多相流快速模拟*

刘欢¹, 刘雪梅^{1,2}, 郭松³

(1. 华北水利水电学院 信息工程学院, 郑州 450011; 2. 北京航空航天大学 自动化科学与电气工程学院, 北京 100191; 3. 呼伦贝尔学院 传媒学院, 内蒙古 海拉尔 021008)

摘要: 对于流体和其他物体的交互, 提出了一种基于 Lattice Boltzmann 的建模和绘制方法。针对固、液交互提出了外力叠加机制, 考虑了障碍物对液体的单向作用; 在液、液交互时考虑了两种液体的相互作用力。采用 GPU 硬件加速技术对 LBM 算法进行了加速, 并采用基于屏幕空间的绘制技术对流体表面进行了绘制。实现了两种不相溶液体交互, 以及液体与固体交互场景的模拟。

关键词: 格子波尔兹曼方法; 多相流; 硬件加速; 屏幕空间绘制

中图分类号: TP391.9 **文献标志码:** A **文章编号:** 1001-3695(2013)05-1564-04

doi: 10.3969/j.issn.1001-3695.2013.05.074

Fast 3D Lattice Boltzmann simulation of multiphase

LIU Huan¹, LIU Xue-mei^{1,2}, GUO Song³

(1. Dept. of Information Engineering, North China University of Water Resources & Electric Power, Zhengzhou 450011, China; 2. School of Automation Science & Electrical Engineering, Beijing University of Aeronautics & Astronautics, Beijing 100191, China; 3. Institute of Media, Hulunbuir College, Hailar Inner Mongolia 021008, China)

Abstract: This paper presented a method of Lattice Boltzmann based fluid simulation in computer animation. It adopted external stack mechanism with considering the force exerted by the obstacle and the cohesion force between particles, respectively for the interaction between fluid and solid obstacle, and one fluid interaction with the other fluid. Then, it accelerated the algorithm with graphics process unit and rendered the fluid based the screen space method. Finally, it simulated the scene of fluid interaction with some other thing.

Key words: Lattice Boltzmann method(LBM); multiphase flow; acceleration; screen space method

流体模拟主要是指结合流体模拟的物理现象、方程和计算机图形学的方法来模拟海面、海浪、烟雾、火焰等场景, 是计算机图形学中的重要研究内容, 其在影视特技、电脑游戏等领域有着广泛的用途。由于人们对电影、游戏的视觉效果要求越来越高, 观众希望能够看到真实流体流动的效果。于是, 基于物理的绘制方法逐渐广泛流行。目前, 基于计算流体力学, 通过求解 Navier-Stokes (N-S) 方程的模拟方法国内外已经有了大量的研究成果。格子波尔兹曼方法(LBM)是近二十年来快速发展起来的一种非常有吸引力的计算方法, 它为流体动力学提供了另一种表达方式。由于其容易处理流体内部的相互作用, 适合处理复杂几何构形流动; 计算具有局部性和并发性, 易于并行, 因此被广泛用于模拟流体, 如单向流、多相流、多孔介质流、微观流动等。

当前基于 LBM 的模拟方法主要集中在数值运算方面, 通常是基于二维的 LBM 模型或对三维流场中的某个剖面的物理量进行计算, 运用 LBM 进行三维动画场景的模拟^[1]还比较少见。对于液体与液体交互, 由于流体系统中增减一种流体, 问题的复杂性并不仅仅只增加了一倍, 除了两种流体自身内部的相互作用之外, 两种不同流体之间也存在相互作用, 并且这种相互作用通常很复杂^[2], 因此模拟比较费时。

鉴于 LBM 方法的并行性, GPU 及集群技术可用来进行流体模拟时的加速。文献[3]对基本 LBM 算法进行了实现; 文献[4]研究了 GPU 模拟单相流、多相流加速的可行性; 文献[5]对并行实现进行了优化, 探讨了存储器访问的优化技术; 文献[6]对氧气和氮气混合扩散的问题利用二维 LBM 进行了分析, 其实现基于 CUDA 进行仿真。流体的表面绘制是流体模拟的关键, Thürey 等人^[7]基于 LBM 自由表面方法结合 level set 来计算水体表面曲面和模拟较薄的水。Bhattacharya 等人^[8]利用 level set 优化薄板能量的方法构造流体表面, 绘制能够表现流体表面细节, 但是由于涉及预测速度和保证质量守恒, 因此很难用于实时绘制。Muller 等人^[9]提出了一种基于屏幕空间的绘制方法, 该方法只绘制由视点决定的表面一层粒子, 绘制速率较快。

本文为了实现三维场景的模拟, 采用了三维 LBM 的 D3Q19 模型, 对于多相流之间的相互作用, 选取 Shan-Chen 模型, 由于其只计算邻近一层格点的作用, 从而保持了 LBM 的局部计算和易于实现的优势。

本文的主要贡献是, 基于 Shan-Chen 模型, 提出了固体和液体交互的外力叠加机制; 采取 D3Q19 模型的 LBM 计算部分, 利用 CUDA 进行加速, 并采用基于屏幕空间的方法对流体

收稿日期: 2012-08-30; **修回日期:** 2012-10-10 **基金项目:** 国家自然科学基金重大项目(61190121); 郑州市科技攻关基金资助项目(112PPTGY222)

作者简介: 刘欢(1988-), 女, 河南信阳人, 硕士研究生, 主要研究方向为图形图像处理、流体模拟(lj. lh@163.com); 刘雪梅(1965-), 女, 教授, 博士, 主要研究方向为计算机图形学、虚拟现实; 郭松(1981-), 男, 内蒙古四子王旗人, 讲师, 主要研究方向为图形图像处理。

进行了绘制。基于 LBM 实现了三维多相流场景的模拟。

1 LBM

LBM 是一种较新颖的求解 Navier-Stokes 方程的数值方法。和传统的 CFD (computational fluid dynamics) 方法着眼于宏观物理量不同, LBM 基于微观模型和力学方程 (Lattice Boltzmann 方程)。该方法的基本思路是从微观分子的角度建立流体运动方程。从平均意义上讲, 该方程描述的物理量在宏观角度下满足 Navier-Stokes 方程。

1.1 格子 BGK 模型

单松弛时间 Boltzmann 演化方程可表示为

$$f_i(x + c_i \Delta t, t + \Delta t) - f_i(x, t) = -\frac{1}{\tau} [f_i(x, t) - f_i^{eq}(x, t)] \quad (1)$$

其中: $i = 0, 1, \dots, N$ 为每个格点不同的离散速度方向; τ 是无量纲松弛时间; c_i 是离散速度; f_i 是粒子分布函数; f_i^{eq} 是平衡态分布函数。

1992 年, Qian 等人提出 DnQb 模型 (n 代表空间维数, b 是离散速度数), 这些模型到目前为止仍然应用最为广泛。本文将选 D3Q19 模型进行介绍。D3Q19 模型的离散速度为

$$c_i = \begin{cases} (0, 0, 0)c, & i = 0 \\ (\pm 1, 0, 0)c, (0, \pm 1, 0)c, (0, 0, \pm 1)c & i = 1, 2, \dots, 6 \\ (\pm 1, \pm 1, 0)c, (\pm 1, 0, \pm 1)c, (0, \pm 1, \pm 1)c & i = 7, 8, \dots, 18 \end{cases} \quad (2)$$

其中: $c = \Delta x / \Delta t$ 为格子速度, Δx 和 Δt 分别为格子步长和时间步长。

其平衡态分布函数 f_i^{eq} 为

$$f_i^{eq} = w_i \rho \left[1 + \frac{(c_i \times u^{eq})^2}{c_s^2} + \frac{(c_i \times u^{eq})^2}{2c_s^4} - \frac{(u^{eq} \times u^{eq})}{2c_s^2} \right] \quad (3)$$

其中: $c_s^2 = \frac{1}{3}c^2$; w_i 是权重系数, 对于速度的 19 个离散方向上

$$w_i = \begin{cases} 1/3 & i = 0 \\ 1/18 & i = 1, 2, \dots, 6 \\ 1/36 & i = 7, 8, \dots, 18 \end{cases} \quad (4)$$

流体的宏观密度、宏观速度分别为

$$\rho = \sum_i f_i, u = \frac{1}{\rho} \sum_i c_i f_i \quad (5)$$

流体粘性系数和松弛因子之间的关系表示为

$$\nu = (\tau - \frac{1}{2})c_s^2 \Delta t \quad (6)$$

1.2 多相流 LBM 模型

由于 Boltzmann 方法是描述微观粒子之间守恒的特性, 因此能方便地表达不同相之间的相互作用, 粒子之间的相互作用力可表示为^[10]

$$F(x) = -G\varphi(x, t) \sum_i w_i \varphi(x + c_i \Delta t, t) c_i \quad (7)$$

其中: G 为两相液体相互作用强度, 可以通过调整 G 来改变流体的表面张力; $\varphi(x, t)$ 是相互作用势能, 为了满足等温过程, 可采用如下形式的相互作用势能:

$$\varphi(x, t) = \varphi_0 e^{-\rho_0/\rho} \quad (8)$$

此时的状态方程为

$$p = \rho RT + \frac{1}{2} GRT \varphi^2(x, t) \quad (9)$$

其中, $RT = c_s^2 = \frac{1}{3}$, 式(9)的第一项代表理想气体, 第二项是非线性部分。

流体与固体之间的作用力为

$$F_{ads}(x) = -G_{ads} \varphi(x, t) \sum_i w_i s(x + c_i \Delta t, t) c_i \quad (10)$$

其中: G_{ads} 是吸附参数, G_{ads} 可以调整液体在固体表面接触角的大小。当 $x + e_i \Delta t$ 是流体所在位置时, s 为 0; 当 $x + e_i \Delta t$ 为固体所在位置时, s 为 1。

根据文献[11], 粘性关系 G 和吸附参数 G_{ads} 之间的关系为

$$G_{ads} = G\varphi(\rho(x, t)) \quad (11)$$

重力的计算公式为

$$F_g(x, t) = \rho(x, t)g \quad (12)$$

其中: g 为重力加速度。

LBM 方法中引入力的方法有多种, 原始的 Shan-Chen 模型通过改变平衡态分布函数 (式(3)) 中的平衡速度来反映粒子之间相互作用的影响, 因此平衡速度为 $u^{eq} = u + \tau F / \rho$, 流体的真实速度为 $U = u + F / (2\rho)$ 。本文考虑的流体与固体交互的模拟中, 体积力由粒子间的相互作用力、液体和固体之间的相互作用力 $F_{ads}(x, t)$ 、液体所受的重力 F_g 三个部分组成。

$$F = F(x, t) + F_{ads}(x, t) + F_g \quad (13)$$

1.3 边界条件处理

LBM 的优势在于容易处理边界。常用的边界条件有周期边界条件、无滑移边界条件和自由滑移边界条件。周期边界条件用来模拟边界无限的情况; 无滑移边界常用来模拟阻力很大的障碍边界, 通常使用标准反弹规则; 自由滑移边界条件常用来模拟光滑的障碍边界, 通常采用类似光线反射的规则^[12]。

由于本文处理的是静止固体边界, 因此对边界上的粒子进行弹回处理, 利用标准反弹格式。粒子在碰撞和迁移过程中, 碰撞后从流体向壁面迁移的粒子速度是已知的, 而从壁面向流体迁移的粒子速度未知。为了保证在壁面处流体粒子的速度为 0, 假设在边界内部有大小相同、方向相反的假想粒子分布。在迁移过程中, 固壁内的假想粒子迁移至流体之中。图 1 描述了 D2Q9 模型中的反弹边界处理。边界节点未知分布函数可由相应节点的分布函数经过反弹获得, 即

$$f_{2,5,6} = f_{4,7,8} \quad (14)$$

基于 LBM 多相流的具体计算步骤如下:

a) 初始化密度场、速度场, 由式(3)计算 $f_i^{eq}(x, t)$ 。

b) 由平衡分布函数初始化分布函数。

c) 在时刻 t , 根据式(1)右边项执行碰撞过程, 发生改变的节点 x 上的粒子分布函数为原有分布和碰撞项之和, 在此根据式(13)计算体积力。

d) 边界处理。

e) 计算式(1)左边项执行迁移, 节点 x 上的粒子以速度 c_i 运动到节点 x 相邻的节点 $x_i + c_i \Delta t$, 粒子分布函数也随之迁移。

f) 根据式(5)计算时刻 $t + 1$ 时刻的密度和速度。

g) 重复步骤 c) ~ f), 直到条件终止。

2 基于 GPU 的 LBM 多相流的实现

自 2007 年 NVIDIA 公司正式发布了 GPU 并行开发环境 CUDA (compute unified device architecture) 以来, CUDA 和支持 CUDA 的 GPU 性能不断提高, 功能不断完善。到目前为止, CUDA 版本已经进行了 4 次提升, 功能上有很大改进, 可以很好地支持硬件的新特性。

CUDA 编程模型将 CPU 作为主机, 将 GPU 作为设备协同工作。CPU 负责进行逻辑性强的事务处理和串行加速, GPU 负责执行高度线程化的并行处理任务。一个系统中可以存在一个主机和多个设备。

由于一个 GPU 上面可以同时执行数百个线程, 因此通过一个线程处理一个节点上的分布函数的迁移和碰撞就可以实

现并行加速。

本文的 Boltzmann 方程的数值计算部分在单个 GPU 的实现过程分为以下四个步骤:

a) 在 CPU 和 GPU 上分配内存。在设备端分配 19 个一维数组 ($d_{f0}, d_{f1_old}, \dots, d_{f18_old}$) 存储当前时间步上的 19 个方向上的速度分布函数, 每个数组的长度为 $NX \times NY \times NZ$; 由于 $c_i = (0, 0, 0)$ 的分布函数并不迁移, 因此分配了 18 个一维数组 ($d_{f1_new}, d_{f2_new}, \dots, d_{f18_new}$) 存储下一时刻 18 个方向上的速度分布函数。速度分布函数在奇数时间步由 f^*_old 传播到 f^*_new 中, 在偶数时间步由 f^*_new 传播到 f^*_old 中。在设备端分配 3 个一维数组保存所有格点的 x, y, z 方向的速度。在 host 端分配 19 个一维数组存储速度分布函数, 分配 6 个一维数组 ($h_{uxold}, h_{uyold}, h_{uzold}, h_{ux}, h_{uy}, h_{uz}$) 存储当前时间和下一时间步的速度。

b) 数据传递。初始化数据后将 host 端数据复制到 device 端。

c) 在 GPU 上获得计算结果。在 device 端执行粒子迁移、碰撞、流体与障碍物交互边界、粒子间相互作用力的处理。其中速度分布函数在奇数时间步由 f^*_old 传播到 f^*_new 中, 在偶数时间步由 f^*_new 传播到 f^*_old 中 ($i=0, 1, \dots, 18$)。

d) 一次迭代完成后将结果传递回 host 端。

e) 释放 CPU 和 GPU 上的内存。

为了提高计算效率, 将碰撞步和迁移合并为一个内核函数。

3 表面绘制

以上部分完成了流体模拟的数值计算。对于流体的表面绘制, 采用基于屏幕空间的方法, 其基本原理是输入三维空间中一系列的点集 $x_1, \dots, x_N \in R^3$ (在 LBM 方法中, 这些位置由含有流体的格点决定), 设置投影矩阵, 由视点到世界坐标中粒子的位置确定流体的表面, 如图 2 所示。

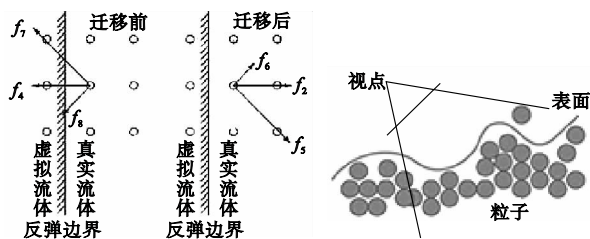


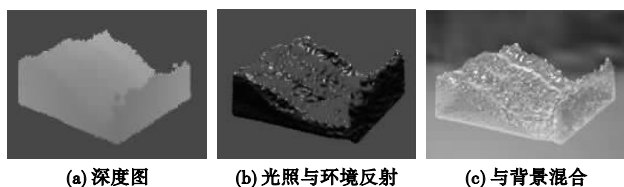
图1 反弹边界示意图

本文中具体实现步骤是:

a) 把 LBM 含有流体的格点处绘制成粒子球, 截取深度缓存获得深度图, 根据格点的密度获得与流体透明度相关的深度图, 然后对厚度图进行平滑处理, 得到的效果如图 3(a) 所示。

b) 将深度图作为流体的表面, 根据平滑后的深度图计算法线, 根据法线计算流体表面的光照和环境反射, 得到的效果如图 3(b) 所示。

c) 将得到的厚度图与背景进行混合, 绘制出流体, 如图 3(c) 所示。



(a) 深度图

(b) 光照与环境反射

(c) 与背景混合

图3 流体绘制

4 实验结果

实验的硬件配置: CPU 为 Intel® Core™ Quad Q9400 (四核)

2.66 GHz, 4 GB 内存, NVIDIA GeForce GTX470 显卡。软件平台为 Windows 7 操作系统。程序编译运行环境为 CUDA 4.0。为了验证并行程序的有效性, 本文在前期实验过程中, 首先选择了 D2Q9 模型中经典算例对顶盖驱动流进行了模拟验证, 并采用 CUDA 进行了加速, 表 1 是实验得到的数据。

表1 CPU 与 GPU 每秒更新格点数对比

网格大小	CPU	CUDA (每秒更新节点数/加速比)			
		32	64	128	256
128 × 128	5.31	275.18/51.82	334.12/62.92	318.16/59.92	350.12/65.94
256 × 256	5.36	387.16/72.23	508.16/94.81	510.30/95.21	496.16/62.57
512 × 512	5.29	478.15/90.39	562.12/106.27	576.56/108.99	598.22/113.09
1024 × 1024	4.53	439.24/96.96	610.66/134.81	616.52/136.10	613.25/135.38
2048 × 2048	4.62	449.26/97.24	628.15/135.96	624.36/135.14	648.74/140.42

由表 1 可以看出, 随着网格规模的增大, GPU 多线程每秒更新的节点数逐渐增加, 相对于 CPU 的加速比也随之增大并趋于稳定; 在计算规模相同时, 当 block 里线程个数为 32 时, 程序加速效率较低, 而当线程数为 64、128、256 时, GPU 的加速比较大。

对于 D2Q9 模型和加速比, 中国科学院过程所李博等人^[13]实现了一种耦合 NVIDIA 和 AMD 两类 GPU 的 LBM 凹槽流模拟的算法, 针对 2048 × 2048 的网格取得了两百多倍的加速比。由于加速比涉及到 CPU 的实现, CPU 程序的快慢直接影响到加速比。实验中的对比说明了 CUDA 加速的有效性。

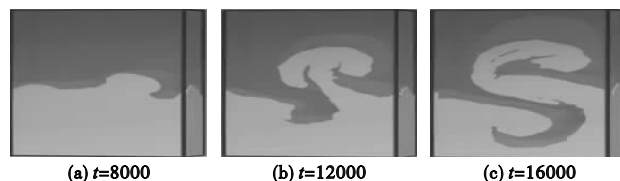
目前, 流体模拟最流行的方法之一——SPH, 由于邻近粒子处理影响计算并行性, 而 LBM 算法碰撞迁移只在局部进行, 使其具有天然的并行性。表 2 是文献[7]中 SPH 方法与本文方法在计算性能上的对比。网格规模是 128 × 128 × 128。

表2 不同方法性能对比

方法	并行程度	平均帧率/fps
LBM	高	20
SPH	中	8

由表 2 可见 LBM 在并行上的优势, 因此, 本文利用 CUDA 对 D3Q19 模型对三维流体进行仿真。

对于互不融合的两液体的交互, 分别用两种颜色代表两种流体, 两种液体分别分布在立方体上下两部分的格点上, 在两种流体边界处格点的密度加上随机扰动, 根据式(7)就可以计算出粒子相互作用力, 根据第 1 章中步骤可模拟出液液两相流的交互过程。在粒子相互作用力的驱动下, 流体开始运动。实验中总的时间步为 20 000, 图 4 为分别在第 8 000、12 000、16 000 时间步时的流体演化的图像。



(a) $t=8000$

(b) $t=12000$

(c) $t=16000$

图4 不相溶液体的交互过程

对于流体与固体的交互, D3Q19 模型选取的网格大小为 $80 \times 40 \times 40$, 设置如 1.3 节中的边界条件, 利用第 3 章中的渲染步骤, 模拟池中水从池子的一端涌向另一端的情景如图 5 所示。

5 结束语

基于物理的流体动画研究一直是计算机图形学的一个研

究热点,随着算法的改进和计算机硬件的发展,不论在真实感上还是模拟的复杂度上都比以前有了很大提高,但是算法速度满足交互程度并不高,关于流体模拟的方法仍需探究。

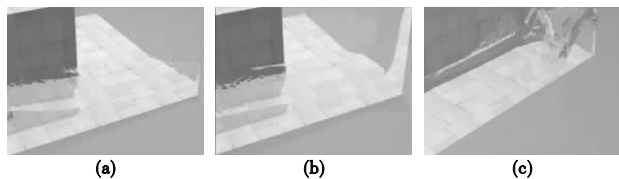


图5 池中水流动效果

本文在 GPU 上对物理计算部分进行了实现,详细介绍了 LBM 利用 CUDA 进行编程的过程;流体的绘制部分采用基于屏幕空间的方法,绘制部分实现了实时;模拟出了多相流交互的场景。本文的缺陷是:针对不互溶的两种液体的交互,两种流体的密度比取值比较小,当密度比大于 10 时,会出现不稳定;另一方面,本文对 LBM 进行了基本的 CUDA 加速,但并没有进行存储器优化、资源均衡优化。因此,设计适合密度比较大的两种液体的融合的算法,把 LBM 的演化步骤进行优化并提高性能,是笔者下一步努力的方向。

参考文献:

- [1] 王国锦,陈雷霆,何明耘. 基于 LBM 模型在 GPU 上实时草波动的实现研究[J]. 计算机应用,2006,26(S2):271-275.
- [2] 朱红斌,刘学慧,柳有权,等. 基于 Lattice Boltzmann 模型的液-液混合流模拟[J]. 计算机学报,2006,29(12):2071-2079.
- [3] KUZNIK F, OBRECHT C, RUSAOUEEN G, *et al.* LBM based flow simulation using GPU computing processor [J]. *Computers and Mathematics with Applications*, 2010, 59(7):2380-2392.
- [4] MYRE J, WALSH S D C, LILJA D, *et al.* Performance analysis of

single-phase, multiphase, and multicomponent Lattice Boltzmann fluid flow simulations on GPU clusters[J]. *Concurrency and computation: Practice and experience*, 2011, 23(4):332-350.

- [5] 黄昌盛,张文欢,侯志敏,等. 基于 CUDA 的格子 Boltzmann 方法: 算法设计与程序优化[J]. 科学通报,2011,56(28-29):2434-2444.
- [6] SAFI M A, ASHRAFIZAADEH M, ASHRAFIZAADEH A A. Lattice Boltzmann simulation of binary mixture diffusion using modern graphics processors [C]//Proc of International Conference on Fluid Mechanics, Heat Transfer and Thermodynamics. 2011:875-881.
- [7] THUREY N, RUDE U. Free surface Lattice-Boltzmann fluid simulations with and without level sets [C]//Proc of Vision, Modeling and Visualization Conference. Amsterdam:IOS Press,2004:199-208.
- [8] BHATTACHARYA H, GAO Yue, BARGTEIL A W. A level-set method for skinning animated particle data [C]//Proc of ACM SIGGRAPH/Euro-graphics Symposium on Computer Animation. New York:ACM,2011:17-24.
- [9] MULLER M, SCHIRM S, DUTHALER S. Screen space meshes [C]//Proc of ACM SIGGRAPH/Euro-graphics symposium on Computer animation. New York:ACM.2007:9-15.
- [10] YUAN P, SCHAEFER L. Equations of state in a Lattice Boltzmann model[J]. *Physics of Fluids*, 2006, 18(4):1-11.
- [11] SUKOP M C, THORNE D T. Lattice Boltzmann modeling: an introduction for geoscientists and engineers[M]. Berlin: Springer-Verlag, 2005:1-11.
- [12] 王长波,张卓鹏,张强,等. 基于 LBM 的自由表面流体真实感绘制[J]. 计算机辅助设计与图形学学报,2011,23(1):104-110.
- [13] 李博,李曦鹏,张云,等. 耦合 NVIDIA/AMD 两类 GPU 的格子玻尔兹曼模拟[J]. 科学通报,2009,54(20):3177-3184.

(上接第 1563 页)了人脸表面几何信息和邻域信息,因此在识别率和等错误率上都体现了一定的优势。

表1 算法结果比较

识别方法	识别率/%	EER/%	识别方法	识别率/%	EER/%
文献[3]	—	5.93	文献[6]	93.78	—
文献[4]	94.67	—	本文	95.5	5.67
文献[5]	—	8.32			

4 结束语

本文提出了一种局部描述符来进行三维人脸识别。该方法首先提取人脸等测地轮廓线并进行重采样,通过分析轮廓线上点的局部点云分布信息和面积特征提取不依赖于坐标信息的局部描述符,最后在测试人脸和库集人脸对应点之间进行局部特征加权求和规则融合,并利用最近邻分类器获得最终识别结果。在 FRGC v2.0 数据库中验证了本文算法的性能,实验结果证明了该方法具有较好的识别性能。同时针对等测地轮廓线在人脸下半部分区域易受表情影响的现象,提出改进措施,实验结果证明了改进方法的有效性。

参考文献:

- [1] 李晓莉,达飞鹏. 基于排除算法的快速三维人脸识别方法[J]. 自动化学报,2010,36(1):153-158.
- [2] WANG Y, LIU J, TANG X. Robust 3D face recognition by local shape difference boosting[J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2010, 32(10):1858-1870.

- [3] PAN G, WU Y J, WU Z H. Investigating profile extracted from range data for 3D face recognition [C]//Proc of IEEE International Conference on Systems, Man and Cybernetics. 2003:1396-1399.
- [4] DRIRA H, AMOR B B, DAOUDI M, *et al.* Pose and expression-invariant 3D face recognition using elastic radial curves [C]//Proc of the British Machine Vision Conference. 2010:1-11.
- [5] WANG Y, PAN G, WU Z, *et al.* Sphere-Spin-Image: a view point-invariant surface representation for 3D face recognition [C] //Proc of International Conference on Computational Science. 2004:427-434.
- [6] AL-OSAIMI F, BENNAMOUN M, MIAN A. Integration of local and global geometrical cues for 3D face recognition[J]. *Pattern Recognition*, 2008, 41(3):1030-1040.
- [7] DORAI C, JAIN A K. Cosmos-a representation scheme for 3D free-form objects [J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 1997, 19(10):1115-1130.
- [8] ZHANG L Y, RAZDAN A, FARIN G, *et al.* 3D face authentication and recognition based on bilateral symmetry analysis [J]. *Visual Computer*, 2006, 22(1):43-55.
- [9] BRONSTEIN A M, BRONSTEIN M M, KIMMEL R. Three-dimensional face recognition [J]. *International Journal of Computer Vision*, 2005, 64(1):5-30.
- [10] DIJKSTRA E W. A note on two problems in connection with graphs [J]. *Numerische Mathematic*, 1959, 1(1):269-271.
- [11] PHILLIPS P J, BOWYER K W, SCRUGGS T, *et al.* Overview of the face recognition grand challenge [C]//Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2005:947-954.